AD-A070 934 OPERATING SYSTEMS INC WOODLAND CALIF F/G 17/2
FTD USER COMMUNICATIONS DEVELOPMENT.(U)
MAY 79 J A BALTER, A TOMLINSON, R V WEEKS F30602-77-C-0220
UNCLASSIFIED OSI-R79-005 RADC-TR-79-94 NL

12

AD A070934

LEVEL II

RADC-TR-79-94
Final Technical Report
May 1979

FTD USER COMMUNICATIONS DEVEOPMENT

Operating Systems, Inc.

J.A. Balter
A. Tomlinson
R.V. Weeks

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DDC FILE COPY

DDC
RECEIVED
JUL 10 1979
A

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 07 09 025

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-94 has been reviewed and is approved for publication.

APPROVED: Edward C. Buzas

EDWARD C. BUZAS
Project Engineer

APPROVED:

ROSS H. ROGERS, Colonel, USAF
Chief, Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAP), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

(19) **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

(18) 1. REPORT NUMBER RADC-TR-79-94	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
(6) 4. TITLE (*and Subtitle*) FTD USER COMMUNICATIONS DEVELOPMENT.		(9) 5. TYPE OF REPORT & PERIOD COVERED Final Technical Report. 27 Sep 77 - 30 Sep 78,
		(14) 6. PERFORMING ORG. REPORT NUMBER OSI-R79-005
(10) 7. AUTHOR(s) J.A./Balter, A./Tomlinson R.V./Weeks		8. CONTRACT OR GRANT NUMBER(s) (15) F30602-77-C-0220 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operating Systems, Inc. 21031 Ventura Boulevard Woodland Hills CA 91364		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64750F (17) 02 (16) 20530207
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRAP) Griffiss AFB NY 13441		(11) 12. REPORT DATE May 1979
		13. NUMBER OF PAGES 41
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*) Same (12) 77p.		15. SECURITY CLASS. (*of this report*) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Edward C. Buzas (IRAP)

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

User Interface	Distributed System
User Language	Intelligent Terminal
Workstation	
DBMS	

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report describes the evolution of the FTD User Communications System concept from an expanded query facility for specific FTD Scientific and Technical Information System (STIS) to a facility running on a minicomputer and supporting a range at computer-based activities, including a generalized data base and file access, text handling and report generation, calculation, simple plots and graphics, and access via direct links to programs and data on large mainframes. → next page

(Cont'd on reverse)

→ next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

391 994

The User Communication System (UCS) design is based on a Digital Equipment Corporation PDP-11/70, to which are attached a number of devices, and peripherals. The report describes the development of a "breadboard" UCS which would initially support two user workstations. Each workstation would provide an intelligent alphanumeric terminal, an interactive graphics display, a function button keyboard for the entry of User Language commands, and a cassette disk for storing personal work files in progress.

A major emphasis in the UC System was the design of a powerful, but simple and easy-to-use language. This language would adjust to the skill level of the user and would provide a unified and integrated means of executing a diversity of functions. In addition, users would be able to utilize the functional building blocks of the language to define their own functions.

The breadboard aspect of the UCS is emphasized in the Evolution Program, which was an integral part of the effort. Groups of potential users would receive orientation sessions in the use of the system. After a period of hands-on experience, these individuals would be able to provide feedback for improving and tuning the UC System.

EVALUATION

The intent of the User Communications Development effort was to increase the intelligence production (quality and productivity of the analyst output) at FTD to meet the increasing mission workload. This was to be accomplished in several ways. These were: a. To improve the man to machine interface by an operating system (with associated peripherals having extensive computing power) able to be used by non-programmer analyst personnel, and, b. To improve the motivation of the analyst to utilize computer power to produce reports quickly and efficiently. This also was to be improved by hands-on experience using the computing power of the PWB/Unix (Programmer's Workbench) breadboard system. This program fits into the RADC Technology Plan (P1A) via the S&T (Scientific and Technical) Data Base Development. It was to be a significant improvement in accessing the data base.

Edward C Buzas

EDWARD C. BUZAS
Project Engineer

Accession For
NTIS GRA&I ☑
DDC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist A | Avail and/or special

CONTENTS

1.0 INTRODUCTION AND SUMMARY

As this report is being prepared and published, significant activities are apparent in both military and civilian intelligence organizations. Large scale efforts are under way to provide improved computer support to intelligence analysts. The U.S. Army is continuing its development of user interface technology for tactical analysts in the field. Rome Air Development Center has responded to the need to provide additional capabilities for Air Force tactical intelligence, operations and C^3 analysts by initiating the multi-faceted 2315 Program. The U.S. Air Force Foreign Technology Division (FTD) is currently planning for upgrades to its ADP facilities.

Intelligence is a phenomenon that is global in scope; moreover, military and civilian intelligence agencies are experiencing an increasing commonality of requirements. Communications networks to support worldwide intelligence data gathering and distribution are being established or improved. Various intelligence entities require information from the others. As a direct consequence of these trends, links to existing communications networks are being installed between the agencies, and new linkages are planned.

An intelligence organization typically supports a number of distinct in-house data bases, and often even multiple DBM systems. Every system has its own query system; the host computer also has its unique operating system control language. When external networks are introduced, new kinds of data-- each potentially having a different format and protocol-- have to be reckoned with. As a result, a user interface problem of serious proportions can develop.

The User Communications System (UCS) described in this report, while currently oriented toward the communications problem internal to FTD, also addresses many of the more general user interface issues. This system was designed to support analyst/users with little or no background in computers. Wherever possible the bewildering profusion of languages and access protocols have been replaced by a single, powerful, easy to use interaction language.

This document is essentially a detailed status report. It is an attempt to provide an account of project accomplishments during the first half of an initially planned two-year effort. Section 2 attempts to summarize the philosophy shared by the User Communications (User Comm) Project team. In Section 3 a sequence of events is described in which the UCS concept evolved: an effort to provide a user interface to a single Scientific and Technical data base (STIS) became a development with much wider applicability across FTD ADP activities. The possible areas of utility of the User Communications System within FTD are investigated in Section 4. Section 5 gives a brief historical account of the project, while aspects of the User Comm effort considered by the project team to be significant contributions are presented in Section 6.

2.0 CONCEPT AND PHILOSOPHY OF FTD USER COMMUNICATIONS

In recent years FTD has faced the challenge of a constantly increasing workload, as well as ever greater variety and sophistication in the analysis of Scientific and Technical intelligence data. Automated support has become an important resource in fulfilling the FTD mission. In some engineering activities ADP help is essential to process incoming raw data; in other activities, analysts may increase their effectiveness many-fold by exploiting the speed, computational power and data handling facilities provided by computers.

At the present time, FTD is experiencing a period of multi-direction expansion in data processing efforts. Numerous development programs have produced special purpose automated systems, each with its own user interface. The User Communications project objective is to provide a breadboard system for an integrated, easy to use, general purpose user interface to some of the automated systems in the FTD community. Ultimately, a production UCS would provide the user access to multiple systems by means of an integrated workstation facility. This approach would provide common data storage and data base access across systems to facilitate data flow throughout the intelligence production process.

2.1 Evolution of the Concept

Initially, the UCS was intended to provide an on-line interface only to the Scientific and Technical Information System (STIS). The User Language under this concept of User Comm would provide a STIS query capability that would supplement what was available by means of IPL (Interactive Processing

Language). The Functional Description produced under the current effort contains as an appendix a Requirements Analysis (RA) document. The RA describes an informal survey conducted in the very early stages of the User Comm effort. It was observed that a number of routine activities of intelligence analysts could profitably be automated, and some of the concepts found in the early versions of the "STIS User Communications" documents reflect these findings.

Since STIS employs relational data structures which can become relatively complex, early system designs included an intelligent graphics terminal attached to the Univac 1110, the STIS host computer; the graphics would provide a visually augmented capability for an analyst to "navigate" his way through the data structures. During this stage of the development, there was a nagging worry to the project team that dependence to a great extent on an already saturated mainframe computer would result in an extremely slow system that would be considered unacceptable by the potential community of users. This worry was later translated into a design which would move the STIS User Communications System to a dedicated minicomputer which would be attached to the Univac system. In this configuration, a user would not be affected by the response of the U-1110 except during an actual query to the STI System. As development proceeded, the feasibility of a more general interface was apparent; hence, the concept was expanded to include FTD data files and data bases in general. The purpose of the FTD UCS evolved to one of ameliorating the fragmentation of programming efforts and proliferation of special user interfaces and user analytical tools, by providing an integrated interface for communication with many FTD automated systems and computerized data bases through a single medium.

The initial survey-based study leading into the current breadboard development effort concentrated on characterizing the analyst's cognitive processing faculties and the design of the automated support to increase analyst intelligence production capabilities. The results of these explorations into analyst work patterns led to the design of an interface with immediate, easy to use tools for the computer-naive user, and sophisticated capabilities to be exploited by the experienced computer user. The present effort has proceeded from general requirements to more specific ones. A recent survey, conducted early in the present effort tends to confirm earlier findings; the UCS design described in this report is thus based on requirements inferred from two surveys. Survey results and their implications are described in greater detail in Section 6.1.

2.2 Automated Workbench Solution

The approach initially recommended development of an analyst workstation with intelligent terminals and graphics capabilities, directly connected to a host mainframe. However, advances in minicomputer technology, as well as overloading of the Univac 1110 mainframe computer, indicated the desirability of including a minicomputer based distributed system to handle both editing and simple processing requirements as well as some data base retrieval capability. This approach would provide a cost effective method of exploiting the massive data processing capabilities of a mainframe system such as the Univac 1110, as well as providing simple to use ad hoc capabilities for the casual user. The ad hoc capabilities were what most appealed to some of the analysts interviewed in the first survey.

The UCS was to serve as an automated workbench for the non-computer oriented analyst, who could substitute an intelligent terminal and user defined functions for pen and calculator, and could insert himself into the report generation process with sophisticated time-saving text editing functions. The text and graphics generated at the terminal would be in machine-readable form, permitting the analyst to assemble a copy of outputs that would closely resemble the format of the final product. Emphasis has been placed on the provision of a total set of tools, including various display media most appropriate for the particular application. Dual screens, graphics output, hardcopy output, and menu selection or typed input permit the user to perform tasks in a manner most suited to and tailored for individual work styles.

The initial hardware configuration included two analyst workstations, each consisting of a textual CRT, a graphics CRT, a dual floppy diskette drive, a function button array, and a data tablet. The alphanumeric CRT screen is divided into three areas. A status area contains the current values for time of day, a terse/verbose mode indicator, the current operational mode, the current filename, and possibly a message from the facility manager. The command/query window is a scrolled area containing a record of the user's recent commands, which may be reviewed or edited. The system response window is a scrolled area in which the system displays short responses to the user's command, including error messages.

The graphics CRT is also divided into three areas. The graphic status display area functions identically to the alphanumeric CRT status window. A menu area contains function names and parameters which may be selected via the data tablet and stylus. An extended response window is used to display large volumes of data or to provide graphic depiction of data or data structures.

The workstation also incorporates a data tablet and stylus, which allow the user to select functions, parameters, field names, or data from the graphics CRT display. The data tablet may also be used for the entering of freehand or traced drawings or graphs. The workstation environment also includes a dual diskette drive enabling the user to save information relating to his work in progress.

2.3 Breadboard Approach

The approach to the UCS development was a phased implementation plan. Prioritization of capabilities to be provided was based on the value of a given function from the user's viewpoint, combined with the complexity of implementation. Implementation strata were defined, with a cohesive set of commands and functions being provided at each level, and with additional levels to be provided as resources permitted. Several editions of the UCS breadboard system were defined, thus phasing in the more complex and sophisticated features of the production system. Emphasis was placed on acceptance by the user community. To this end the strategy included:

1. personal interviews with 35 analysts, and distribution of an in depth survey questionnaire to 81 analysts, analysis of the findings of the survey and incorporation of the results in the UCS design;
2. development of a truly user-oriented breadboard system in cooperative consultation with FTD and RADC representatives; the design goals included flexibility for modification as user experience indicated improvement through change, and extensibility for future enhancement and adaptation to new concurrent projects;

3. design of a well planned user training program with pilot groups of user-analysts gaining experience with the system in a simulated FTD environment, and providing data for evaluation;
4. optimization of the user interface based upon experience with the system and its users, and upon their reactions to the system.

This iterative strategy was to result in a User Communications system having features which analysts would find highly convenient to perform their tasks. Automated support was exploited in the design; the objective was an accompanying increase in analyst productivity.

3.0 UCS AS A STIS INTERFACE

3.1 Initial Orientation to STIS

During the early development under previous contract efforts, the STIS and User Communications efforts were closely related. Section 2 indicated some of the considerations which contributed to a wider scope for the User Comm concept. While the UCS design was moving toward greater generality, and various forms of the FTD 1980-83 ADPE plan were being discussed, events were taking place at FTD which had a bearing on the ongoing STIS development, and consequently on the User Comm effort as well.

3.2 Independence from STIS

As the emphasis at FTD shifted away from STIS, the User Communications effort responded by adapting the User Language and implementation plans toward a more general data management approach. Ultimately, funding for needed enhancements to STIS was withdrawn, and emphasis was placed on adapting general purpose data management systems for the storage and retrieval of scientific and technical information of the kinds stored by STIS.

3.3 Generalization of the UCS Concept

The User Comm effort addressed the general problem of access to information stores on undefined host processors. In light of the variety of on-line capabilities needed by intelligence analysts, and in view of the expanded automated support indicated in the ADPE plan, the UCS design turned toward a more generalized data access. More specifically, the FTD User Communications was aimed to provide a centralized facility for accessing data bases and data files, executing intelligence programs on various processors, and generating reports, both of preliminary and finished intelligence. In addition, provision was made in the design to exploit new information made available on the communications network; this includes document files, messages, requests, etc.

4.0 UCS IN AN EVOLVING FTD CONTEXT

With the changes in orientation and scope concommitant with the present contractual effort, significant redirection was required. Two major forces guiding the changes were an alteration in the hardware approach, changing from the intelligent terminal approach to the minicomputer configuration, and abandonment of the plans to interface to the STIS database. The resultant modified design provides capability in the mainstream of current thinking for the global terminal base envisioned in the FY 80-83 ADPE plan.

4.1 FY 80-83 ADPE Plan

In an effort to upgrade the automated support to FTD internal and external users, FTD considered a concept which calls for the merging of computational resources into a unified facility providing distributed processing and versatile communications access. Communications links to FTD computers would be provided not only internally, but also to intelligence and military data network outside of FTD.

A change of such major proportions in the FTD automated support facilities had implications for all existing developmental efforts, and the UCS had potential as a key element in this approach. In particular, three major impacts were apparent:

1. Considerable amounts of additional intelligence materials would be potentially available to the analyst, and on-line access to these materials would be provided.
2. Finished intelligence could be available on-line to those who had a need for it. In the FTD environment, finished intelligence from one

analyst group might become source information for another.

3. Analysts could be included in a message dissemination system in which the reception and routing functions could be performed by the User Comm interface system.

It was expected that the analyst workstation configuration designed in the UCS effort would provide valuable design data for the global terminal base envisioned in the FY 80-83 concept.

4.2 Related projects

In view of the general interface nature of the UCS effort, it was incumbent on the project team to monitor concepts and the progress of other concurrent development projects and production programs at FTD which were known to require user interfaces.

4.2.1 IPS The Intelligence Processing System (IPS) is oriented toward creation and maintenance of a large integrated data base and the production of finished intelligence products from the data base. Information in the data base consists of characteristics of missile systems, command control systems, and event related information. IPS resides on the Univac 1110 and utilizes both STIS and the EXEC8 file system structure. IPS has been expanded to handle additional weapon system and subsystem information and to facilitate the automated support capabilities of the ICBM system for engineering personnel.

4.2.2 IEAS The Integrated Event Analysis System (IEAS) is designed to assist in the analytical processing of ICBM event data. The objective of the analysis is to determine if anything significant occurred during any given event. IEAS also utilizes information in the STIS data base residing on the U1110.

4.2.3 $\underline{C^3}$ ETAS The Command, Control and Communications (C^3) Electromagnetic Threat Analysis Subsystem (ETAS) assists the analyst in determining the battle strategy management scheme any enemy or potential enemy will employ in the event of crisis and hostilities. Information concerning the mix of weapons to be used by the enemy, the doctrine and tactics, target selection, deployment and rules of engagement are, to a large extent, predicted by the C^3 system. The system is being implemented on a DEC* PDP 11/70 utilizing the DBMS-11 database management system.

4.2.4 EW ETAS The Electronic Warfare (EW) ETAS project is proceeding in parallel with the C^3 project described above. Both efforts will demonstrate a capability on the PDP 11/70; the emphasis in the EW development is the design and conversion of analytical algorithms from the U-1110 on the PDP 11. This project has data base access requirements, as well as text editing, report generation, and data analysis and reduction. A desk-top calculator capability is also included. Some information will be communicated from the U-1110 by magnetic tape.

* DEC and PDP are registered trademarks of Digital Equipment Corporation.

PRECEDING PAGE BLANK-NOT FILMED

5.0 PROGRESS TOWARD USER COMMUNICATIONS DEVELOPMENT

The present document reflects the status of the User Communications project after one year of a projected two-year development effort. Approaches taken to the project Statement of Work (SOW) and resulting efforts relating to the SOW tasks are the subject of the following section. Work on certain tasks has not been reported previously except in Monthly Project Status Reports (CDRL Line Item A001).

5.1 Design Document Revision

Task 4.1.1 of the SOW calls for revision of design documents produced under the previous User Comm effort. These documents are the Functional Description (A004), the System/Subsystem Specification (A005), and the Program Specification (A006).

Significant changes in the User Comm scope and approach to system implementation were discussed in Sections 2 and 3. In particular, major portions of the software would reside on a PDP-11 minicomputer rather than on the Univac 1110 as originally planned. As a direct consequence of these changes, the design concept under the previous User Comm effort was greatly modified. For the most part the previous design documents were obsolete, and the documentation effort consisted of complete re-writing rather than revision. Under the Statement of Work the revision was to result from specified investigations:

1. Review of the FTD FY80-83 ADP Concept (Subtask 4.1.1.1),
2. Review of other FTD programs in which the user interface problem is relevant (Subtask 4.1.1.2).

In the course of the program other events occurred which also impacted the design documents; the more important of these were the decision not to use STIS, the decision not to interface to the Univac 1110, and the emergence of the question of what to include in the initial breadboard UCS, given the limited amount of remaining project resources.

5.1.1 FTD FY80-83 ADPE Plan Investigation of the FY80-83 ADP Concept produced no conclusive answers. Early in the effort an unofficial preliminary version of the concept was discussed with FTD and RADC personnel. In this version most FTD processors would be connected together by means of a network. One of the processors in the network would be a DBMS host (this machine would be a replacement for the existing IBM 360 CIRC computer). Accesses to the network would be provided by a Network Control Processor (NCP), a minicomputer of the AN/GYQ-21(V) (PDP-11) type. The NCP would also link to another processor that would provide the interface to external networks (COINS, DIAOLS, etc.). Providing service facilities for FTD users would be another processor called the Global Terminal Base. User terminals attached to this system would gain access via the network to computing resources and to information from the data base(s). A common user language, possibly containing standardized query formats, would facilitate user interaction with all the ADP facilities.

International Computing Company (ICC) was awarded an RADC contract (known as the HARD Project) to study the existing ADPE utilization at FTD and to plan for coordination of RADC contracts at FTD. It was hoped that ICC personnel, working together with FTD planners, would be able to formulate a compatible ADPE plan for FTD in the FY 80 to 83 time frame.

On 2 November 1977 a meeting was held at the Dayton office of ICC with representatives from RADC and ICC, as well as from the User Comm Project. There was considerable discussion of User Comm objectives and ICC activities. It seemed clear, however, that there was a problem of phasing between the two efforts and that the ICC effort would not be in a position to make definitive statements on the FTD ADPE Plan for at least several months from that time.

On February 13 to 15, 1978, a meeting was held at RADC to investigate the possiblility of closer coordination among the FTD programs. In addition to RADC and FTD personnel, the IPS, EW-ETAS, C^3-ETAS, User Comm and HARD (ICC) projects were represented. The FTD ADPE Concept was discussed as one of the issues. Doubt was expressed as to whether a plan yet existed, and if it did exist, whether all user terminals would be supported through a global terminal base.

During the course of the project, and particularly during the period of preparing the UCS design documents, User Comm team members were able to determine neither an official version of the FTD ADP Plan nor a precise definition of long range ADP policy at FTD. We believe, however, that the technology being developed within the User Comm effort is relevant to FTD needs as we understand them.

5.1.2 Related Projects Attempts to coordinate User Comm development with directions being taken by other FTD projects and to provide User Comm support for certain of these projects were relatively unsuccessful.

During the previous User Comm contract, the FTD projects most relevant to User Comm were the Intelligence Processing System (IPS) and the Integrated

Event Analysis System (IEAS), both of which used the STIS information base. If User Comm were eventually to support a variety of users, it would be necessary to find out what kinds of terminal displays, what auxiliary file access and what specialized computer programs were being developed within these projects. Obtaining information concerning IPS was not a problem, since one of the project subcontractors was also involved with IPS. The User Comm COTR provided what information was available on IEAS. A mechanism was set up for the various contractors to exchange monthly status reports. Keeping abreast of STIS development was also not difficult because of contractual relationships between the STIS and User Comm programs. The User Comm team attempted to incorporate into the UCS design, where applicable, information gained from these other projects. The Project Status Report for December 1977 states, "Work continued on the project-internal Requirements Analysis document. Inputs from various sources were incorporated, including information from other FTD projects, where available." Much of the relevance of IPS, IEAS and STIS itself to the UCS evaporated when FTD revised its policy concerning STIS.

Because the EW and C^3 ETAS projects were initiated within a short time of the beginning of the User Comm effort, there was some hope that the three projects might coordinate their efforts to some extent and thus begin to move in the same direction. At the previously mentioned RADC meeting in February, 1978, there was considerable discussion of the issues and problems of such coordination. Schedule conflicts appeared to be a major obstacle to meaningful meshing of the programs. Also brought up was the apparent incompatibility of PDP-11 operating systems between UCS and the two ETAS systems. User Comm project team members made the point that interconnecting (i. e., networking) minicomputers having diverse functions, and possibly different operating systems, is in complete harmony with the modern concept of distributed computer systems.

As long as plans were in effect to interface between UCS and STIS, there was a problem of crucial importance; no funds had been allocated for developing the interface software. Specifications for the interface and scoping of the development effort were completed by one of the User Comm subcontractors, but implementation did not occur.

5.1.3 Impact of the STIS Phaseout Data management capabilities for the UCS were initially planned to be provided through STIS. Preliminary UCS design incorporated these plans in two ways:

1. A unique STIS interface was provided for in the design,
2. Special user-oriented displays, including graphics displays, were envisioned to improve the ease of visualizing and understanding STIS data structures.

Complementing STIS, a more general DBMS query capability was planned as part of the User Language for UCS.

When STIS was eliminated, at least temporarily, as the UCS data base management resource, the system design required revision--not because the previous design was hard-wired to STIS, but because a specific DBMS system or systems needed to be identified to take the place of STIS. Lacking such a system (FTD at that time had not decided on a replacement system for STIS) the project team chose a more general approach. The ADAPT system, which will be described in greater detail in Section 6.2, was investigated as a means to providing a query translation capability for several or many DBM systems.

Much of the DBMS interface problem was already solved in ADAPT. The UCS design documents, including the Functional Description, which had already been released in draft form, required revision to include compatibility with existing ADAPT software and to provide compatibility with part of the ADAPT query language.

5.1.4 Postponement of the Univac 1110 Interface Much of the design for the hardware and software interface between the PDP-11 minicomputer and the Univac 1110 host was completed early in the current effort. For practical reasons this interface was planned as a terminal link between the U-1110 terminal communications equipment and a synchronous communications multiplexor on the PDP-11. Interface mechanisms were described in preliminary form in the Functional Description developed under this contract and were specified in some detail in the draft version of the System/Subsystem Specification. Before the SS could be delivered, the project was advised that FTD would not have space for the UCS hardware during the period of the contract. Although the interface specifications were to remain in the design documents for possible implementation at a later time, there was an impact on the document preparation process. The project team shifted its emphasis at that point to designing the right kind of interface between two processors, i.e., a Direct Memory Access interface supporting higher data transfer rates.

5.1.5 Development Tradeoffs Because of the need (initially) to expend project resources in specifying the STIS interface, the project team attempted to budget remaining resources so as to assure implementation of a breadboard UCS having at least some capabilities in the entire spectrum of planned user functions. Developmental priorities were established for the various capabilities that could be provided. Successive versions of the breadboard system design were considered, and each version was evaluated in terms of implementation effort. The result was incremental design of multiple breadboard systems, each providing additional capabilities over its predecessor and each being described in the design documents.

5.2 Breadboard Development

Development and implementation of a user communications breadboard system is the subject of project Statement of Work Task 4.1.2; originally the system was to be demonstrated on the FTD Univac 1110. The language of the SOW indicates that development should take place in a manner that enhances the opportunity to evaluate the resulting breadboard system. Two subtasks are defined under this activity. The first involves selecting user language commands and system/user functions, while the second has to do with developing specifications for terminals and other hardware needed to support the interface.

Revising and rewriting the design documents and developing the system proceeded in parallel. The previous section details some of the issues and considerations that impacted development.

5.2.1 Defining User/System Functions and Capabilities

Initially because of limited project resources, and later in response to events that affected UCS development, several passes were made on possible stages or phases of implementation for the breadboard system. Always the project objective was to provide a set of capabilities in as many functional areas as possible. As a result of meetings, the first held at the contractor's facility on June 20 to 22 and the second at RADC July 6 and 7, 1978, in which RADC and FTD representatives informed the project team of changes at FTD that would have a direct bearing on the UCS development, an assessment was made of the effort required to develop various capabilities and the sequence in which it was possible to implement them.

Based on these facts, several members of the project team met at the contractor's facility in Dayton, Ohio, on September 19 to 22, 1978, and engaged in detailed technical planning for the breadboard implementation. As shown in Figure 5.2-1, UCS implementation is planned as three packages. Package I contains a considerable amount of system-level programming, which was fundamental to later functions. It includes the development of essential UCS routines, as well as low level interfaces between existing system software and new routines. Also included are firmware development for the microprocessor in the Beehive intelligent terminal, software development for the graphics processor, and adaptation of parts of existing software packages (e.g., ADAPT) to UCS functions. Package II contains some enhancements to UC System software and support for a number of new functions. Package III essentially completes the UCS perspective by providing functions (initially to be simulated because of the absence of real interfaces) to access external computer hosts and DBM Systems. Figure 5.2-2 presents a schedule for implementation of these packages in the context of the overall project plan, including the Evaluation Program.

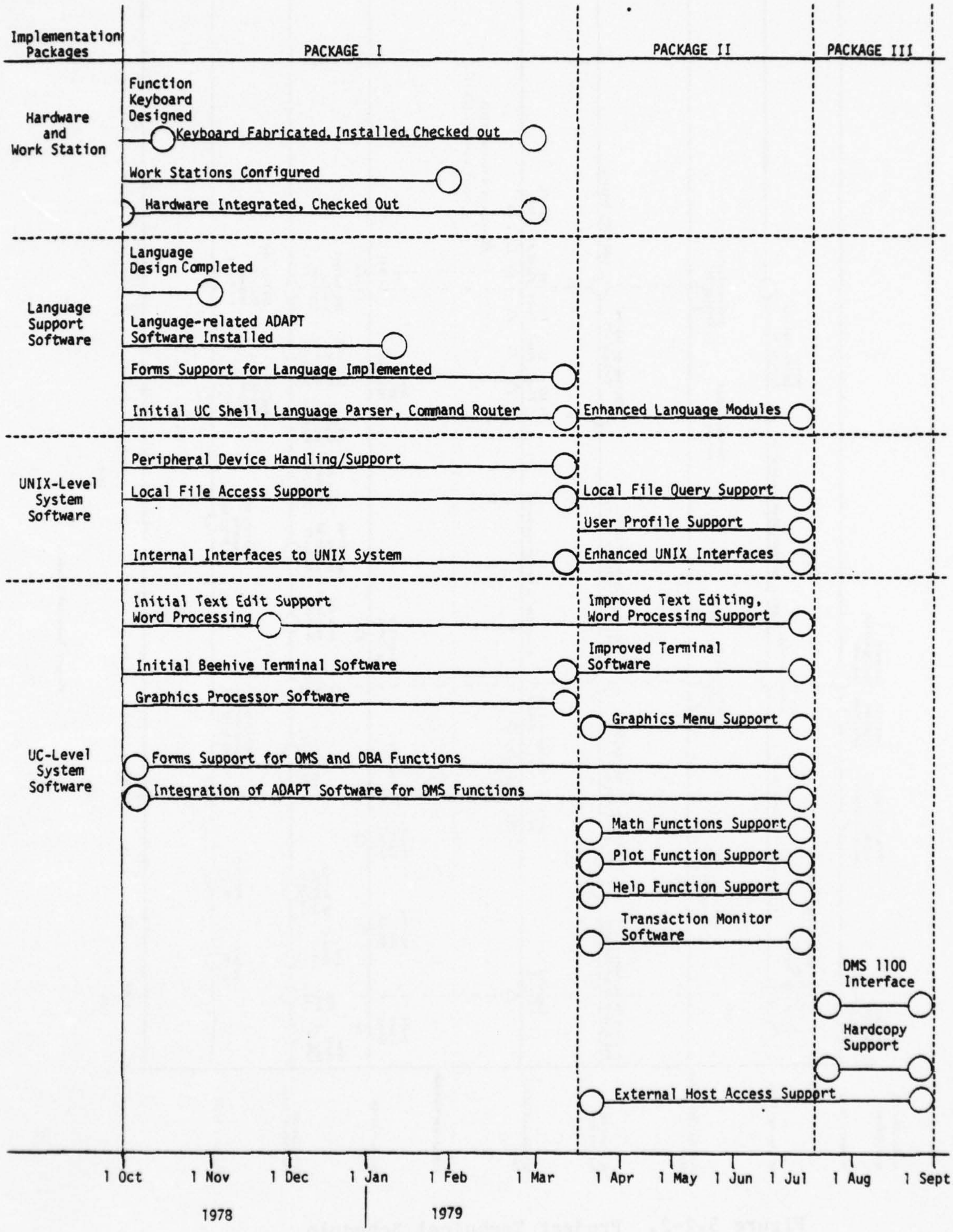
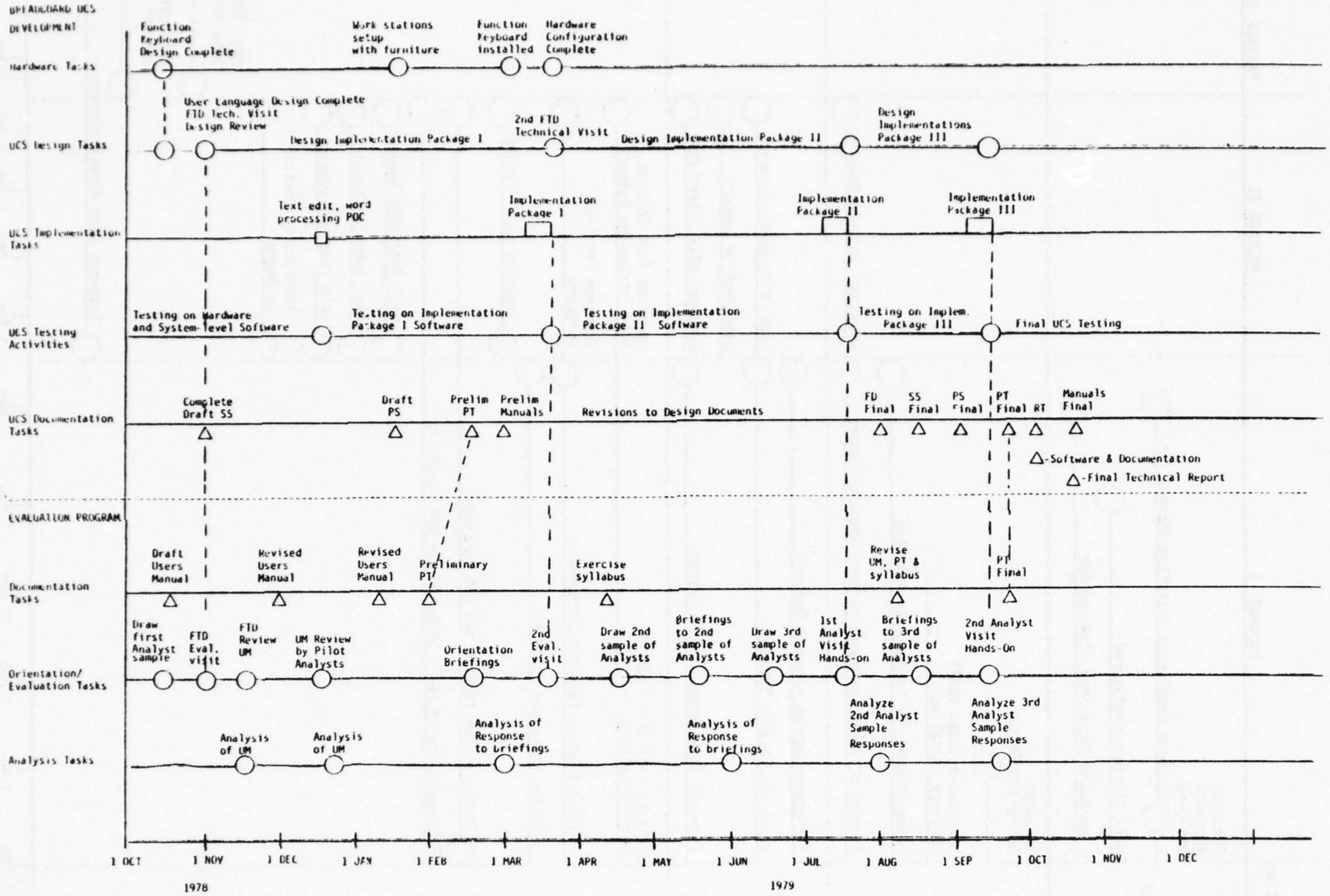


Figure 5.2-1. UCS Implementation Plan



Project Technical Schedule

Figure 5.2-2. Project Technical Schedule

5.2.2 Defining User Language Commands The User Language development consisted of two concurrent distinct, but related activities -- specifying the syntactic form of the language and specifying the set of commands to be included in the repertoire. The syntactic definition was a continuation of the effort initiated under the previous User Comm contract. User Language features are presented in Section 6.2. Language syntax required very careful definition in order to maintain high usability, while at the same time provide an integrated system of commands across a wide range of functions.

Once the general form of the language was designed, choosing the set of commands was a relatively simple matter, consisting of matching user (and system) functions to be provided with command sequences that cause functions and sub-functions to be executed. There were also a number of issues that are more mechanical in nature, including choice and order of command parameters, interactions between various commands, dependencies between commands and physical keyboards, CRT menus, considerations of form fill-ins for parameters, etc.

5.3 Evaluation Program

The User Communications System was originally intended to undergo extensive evaluation, as indicated in the Statement of Work:

Task 4.1.3 Design and implement an evaluation program for the User Communications breadboard system.

Subtask 4.1.3.1 Prepare a preliminary design specifying the concept and objectives of the evaluation plan.

Subtask 4.1.3..2 Provide a global plan of the evaluation program, specifying the number of pilot groups in the program, the level of User Communication implementation to be evaluated, and the kinds of training and orientation to be included in the program.

Subtask 4.1.3.3 Develop a detailed plan for the evaluation program with specific objectives and orientations for the ongoing phases of User Communication System implementation.

Subtask 4.1.3.4 Plan the implementation of the evaluation program. Plan for the selection of pilot groups. Plan for the preparation of user orientation and training materials, including the preliminary copy of the Users and Operators Manual.

Subtask 4.1.3.5 Implement the evaluation program by bringing pilot groups into functional interface with the FTD User Communications System at various implementation stages of the ongoing system development.

Subtask 4.1.3.6 Organize a test and evaluation effort to obtain from trained user groups reactions and responses relative to the facilities and capabilities provided by the User Communication breadboard system.

Task 4.1.4 Analyze the results of the evaluation program and revise the functional, system/subsystem and program specifications developed in Task 4.1.1 for the User Communications System, and the Users and Operators Manual.

5.3.1 Initial Project Activities in the Evaluation Program Milestones for these tasks and subtasks were included in the User Comm project planning documents from the beginning of the effort. As a preliminary to identifying

FTD analysts for eventual participation in the evaluation program, a questionnaire survey was conducted by User Comm team members with the assistance of FTD personnel between December 1977 and February 1978. The survey, the results of which are presented in Section 6.1, also constituted a mechanism for verifying UCS requirements. The program was viewed as a User/System Adaptation effort. User training and orientation would adapt users to experience the system in a hands-on manner; feedback from the users would result in system adaptation and tuning. Iterations of this process would produce a system with a high degree of usability and user acceptance.

5.3.2 Adjustments to the Evaluation Program When it was determined that the UCS would not be installed at FTD according to the original schedule (i.e., the February to April 1979 time frame), discussions were held with the Project COTR and with the Contracting Officer for necessary modifications to the contract Statement of Work. FTD personnel presented a tentative plan in which the UC System would be evaluated at the contractor's site by means of visits from FTD analysts and technical personnel who were to participate in the program. The User Comm project responded with the following draft plan for the evaluation:

Since the activities of evaluation are dependent upon orientation activities as prerequisites, and since the two activities are interrelated at the operational level, it is important to distinguish between their respective purposes and goals. Together they will motivate a sequence of thirteen tasks to be completed during the balance of the current project.

The purpose of user-evaluation activities is to make valid predictions about reactions of analysts to the User Comm system if it were to be installed at FTD in the future. The strategy is to gather reactions from a sample of a analysts to the developmental system, and use these as predictions of reactions to a future system similarly constituted. The validity of such predictions depends upon the care with which the sample is drawn in terms of representativeness of FUTURE universes of FTD analyst-users, and the veridicality of the experimental testing situation in re-creating the important factors that will operate in the future usage situations. These considerations must be kept in mind in designing the evaluation activities.

Several objectives serve the evaluation purpose. The first is to observe analyst-users operating the system, in order to identify weak points that can be corrected during the development phase or identified for future fixes. The second is to secure a full range of OPINIONS from analysts as to good and bad points they perceive about the system and its modes of use, again in order to identify items that can be corrected either during the project or at some future time. The third is to obtain from the analysts suggestions about the areas of perceived future usefulness of such a system for supporting their activities. This data will prove valuable for introducing the system to the wider community at FTD.

The purpose of the orientation tasks is to PREPARE the samples of analysts used in the evaluation so that their responses will represent as valid as possible a prediction of responses of analysts to a future system when a "steady state" has been reached between analysts and system. Several objectives serve this orientation purpose. The first is to refine and improve the orientation materials to the maximum extent feasible before using them to prepare the EVALUATION sample of analysts. Pilot samples of analysts will be used for this purpose. Responses from the pilot samples will be used not only to improve and fine-tune the orientation materials, but also to identify weak points in the system format/repertoire that can be corrected before the final evaluation sample is run. The second is to control the exposure of analyst samples to User Comm so that data on responses to User Comm as a function of number of experiences with the system can be gathered. This objective will require that we seek analyst cooperation in not communicating about User Comm BETWEEN different experimental groups during the period of the orientation/evaluation experiment.

The following sequence of tasks supports the purposes and objectives just discussed:

1. Specify selection criteria for drawing analyst samples.

2. Produce initial draft of user manual.

3. Review initial draft at FTD; (review by FTD User Comm Tech team).

4. Revise manual per FTD feedback.

5. Submit revised draft to pilot sample of analysts, use combination of free-response and probe interviews to gather data on their responses to the manual.

6. Revise manual per pilot sample analyst feedback.

7. From manual, produce materials for four (4) thirty-minute orientation briefings to supplement and introduce manual.

8. Produce exercise syllabus specifying conditions and materials for six (6) coached hands-on online sessions covering main skill and usage

areas of the breadboard UCS.

9. Run sequence of Briefings, Study of the Manual, and Hands-on sessions with a second pilot sample INDEPENDENT of the first. (If time and resources permit, RERUN the first pilot sample on task 9 to provide comparison data and gather additional feedback on "deeper experience" patterns.

10. Revise all materials (manual, briefing materials, hands-on syllabus) per feedback from task 9.

11. Run final larger EVALUATION sample of analysts (INDEPENDENT of both the pilot samples) on the materials revised in task 10. Gather data on: Levels of performance reached as a function of the orientation and training sessions; Range of individual differences experienced; Amount of practice/re-orientation/study need to bring analysts to "practically useful" levels of functioning; System usefulness judgements of analysts as a function of increasing levels of experience with the system; Analyst observations about strengths and weaknesses of the system as a function of increasing levels of experience; Analysts suggestions about future directions in which the User Comm system should be developed (priorities).

12. Document the results of the orientation/evaluation effort, submit draft to COTR.

13. Submit document revised per comments of COTR.

6.0 PRODUCTS OF THE USER COMMUNICATIONS EFFORT

Although the User Communications project has not been carried to completion, and implementation is still in the initial stages, several significant accomplishments can be noted. A survey has been conducted and the results processed; as a consequence, there now exists an up-to-date perspective of activity patterns, clerical abilities and computer-related skills within selected groups of FTD analysts. Whereas the user interface, and in particular the user language, is glossed over as being trivial in the design of too many computer-based systems, this project made language design the crux of the UCS development process. Valuable experience was gained in putting together a minicomputer system and matching it to the capabilities of an operating system. Finally, a user interface design was formulated that contains a technological legacy for future FTD developments.

6.1 Profiles of Potential Users

A questionnaire survey of FTD analysts was made to augment and verify the findings of the interview survey conducted approximately three years earlier, which were summarized in the Requirements Analysis appendix of the Functional Description document of March 1978. The questionnaire required about 40 minutes to complete and consisted of six sections:

1. Finding Need Information (13 items).
2. Doing Calculational Work (18 items).
3. Drafting and Composing Reports/Papers/Notes (9 items).

4. Personal Work Area Information Stores (13 items).

5. Overall Balance of Work Activities (6 items).

6. Comments or Suggestions (1 item).

A stratified random sample of analysts were sent the questionnaire, and 86 completed questionnaires were received back, which represents between one-fifth and one-third of the FTD personnel in analyst shops who are potential users of ADP. Results of the survey were reported in a document "PRELIMINARY ANALYSIS OF THE INFORMATION ACTIVITIES PATTERN SURVEY QUESTIONNAIRE TAKEN AT FTD IN FEBRUARY 1978", 15 September, 1978. Additional planned interviews and augmented data analyses were not conducted due to termination of the project.

6.1.1 Summary of Information Gained from the Survey Questions

Some general patterns were:

1. There were one-third as many responses for computer-supported methods of obtaining information as for other methods.

2. One-third of the respondents do calculations in one-third or more of their tasks.

3. There were three times as many always-usually responses for non-computer methods of calculating as for computer-based methods.

4. About 10% of activities involve computer-supported calculational work, and another 20% involve calculations by other methods.

5. Hardly anyone dictates reports/papers/notes. Most do hand-written drafts which they submit to typists. Not many compose reports on typewriters. Not many use text-editing and word processing.

6. The preponderance of materials stored in offices are textual, with graphic materials a distant second place. Most such files are growing or remaining constant -- few are shrinking. The median size of total work area stores is about 22 linear feet, with about 15% reporting very large stores of ninety or more linear feet. The median number of work files is eight.

7. Computational work and making notes and plans were reported as somewhat less time-consuming than other activities.

8. There is a very high correlation between not consulting CIRC reference search technicians and not doing search oneself. That is, there is a large common quadrant of relatively inactive persons for reference searching. A similar correlation exists between not consulting a Data Bases/Files technician and not searching data bases oneself. Again, there is a common relatively inactive group.

9. For active CIRC users, the median percentage of tasks involving calculations is about 5%, for active data base users about 15%, for search-it-yourself data base users 30%. Thus the active groups for reference searching and data base usage are not the same group.

10. Active data base users are somewhat more likely to do composition by using the keyboard. The same applies to active reference searchers using CIRC.

For the do-it-yourself data base searcher group, more:

1. Spend time doing calculational work than the norm.

2. Spend less time making notes than the norm.

3. Are also do-it-yourself CIRC searchers (but not conversely).

4. Are users of graphics programs.

5. Are users of complex capabilities on hand calculators.

6. Are touch typists (50%).

7. Use fewer files (same amount of space) than others.

For the non-calculationally-oriented subset of analysts, more:

1. Use calculators without memory and fancy functions.

2. Search library themselves; they also do more consulting with the librarian.

3. Are consulters of CIRC and Data Base technicians.

4. Sometimes search CIRC themselves.

5. Are users of computer editing facilities.

6. Are touch typists (more than 50%).

For the calculationally oriented subset of analysts, more:

1. Use hand calculators with complex functions.

2. Use graphics programs.

3. Are data base searchers (do-it-yourself variety).

4. Keep mag tapes and cards.

5. Are likely to write their own programs and run programs themselves.

6.1.2 Discussion of Results

6.1.2.1 Survey Data in Relation to Stated User Comm Requirements Somewhere between a fifth and a third of the FTD people in the analyst shops who are potential ADP users were represented in the survey sample contained in the eighty-six survey questionnaires completed and analyzed. Evaluated in the context of the interview survey which was conducted approximately three years ago, (summarized in the Requirements Analysis appendix to the Functional Description document of March 1978 for the User Communications System), the questionnaire survey contained no surprises. That is, the current survey results largely confirm the findings of the previous requirements analysis upon which the User Communications concept and philosophy was based.

6.1.2.1.1 Significant Clustering Two important clusters or groups of analysts are identified from their respective habits and activities. One group, currently a minority, are users of FTD's present computing resources; they consult automated search services, utilize computer data bases and perform some of their calculations with computational assistance.

The larger group or cluster does not use computer aids. Based on whether or not they utilize computers, the two groups are designated as computer active or inactive. In the computer active sub-groups, some already make very extensive use of computer support.

6.1.2.1.2 Groupings By Computer Activities The questionnaires showed that the computer-inactive groups share certain behaviors, in that they:

1. Do not calculate.

2. Do not search references or data bases/files.

3. Do not type.

There are a number of possible explanations for these facts, among which are the following:

1. The maximum performance job descriptions for these particular analysts do not include ever performing computer-supported activities; i.e., in their particular jobs, computers would provide no significant help.

2. Some in this group maintain negative reactions or images with regards to computers and their use.

Possible forms of such reactions are:

1. Fear of incompetence in relation to use of computational assistance.

2. Uncertainty as to whether computers would help with job-related activities.

3. Belief that computers could provide no help, when in fact computer support could be profitably utilized.

An interesting question might be, "Are formal organization and task definitions responsible in and of themselves for the dichotomy of active and inactive subgroups?"

Taking the computer active group as a whole, one might ask whether it is possible to distinguish degrees of activeness. The survey questionnaire indicates that the answer to this question is yes -- that there are large differences in the manner and the extent to which analysts utilize presently available computer support.

6.1.2.1.3 Summary of Groups in the Sample by Computer Activity Level

Totally Inactive Group

As mentioned previously this group may be represented by:

1. Analysts who can expect no help from computers because of the nature of their job definitions.

2. Analysts who do not presently use computers, even though such support would be helpful to them, because of various negative attitudes or images.

Somewhat Active Group

This group obtains help from computers in a minor to moderate extent. Their utilization of computers involves:

1. Computer-aided computations, either by themselves or by others for their benefit;

2. Computer-aided search of reference documents, data files and data

bases to obtain information necessary to their tasks.

Very Active Group

Analysts in this subset perform a significant amount of their work using computer help. The subgroups are:

1. Computer-aided computations;
2. Computer-aided search of reference documents, data files and data bases.

6.1.2.2 Follow-Up of the Surveys in the UC System Evaluation Program The initial draft plan for the User Communications Evaluation Program called for participation by three separate groups of analysts. Each of these test groups was to include analysts from each subset of the inactive and active groups of analysts described above. It was felt desirable to identify the respective groups at FTD in order to obtain their inputs to the Evaluation Program. This would have allowed a quite thorough understanding of the several factors that would have been taken into account in designing orientation, training, and evaluation procedures to promote maximum acceptance and usage of the User Comm facilities.

6.2 The User Language

From the beginning of the User Communications effort, human engineering considerations strongly influenced the philosophy of the User Language design. The User Language was particularly oriented toward FTD personnel who perform their daily tasks primarily or exclusively using manual methods; analysts who already utilize ADP support would, it was assumed, easily adapt to the new language. Conscious effort went into designing a system that would avoid the confusion and frustration users experience on many computer-based systems.

Significant developments during the period of the project influenced the final form of the User Language. These developments and their impact will be discussed in this section.

6.2.1 Design of a Language for Simplicity and Ease of Use

6.2.1.1 Easy to Learn Learnability was designed into the User Language in several ways. Each of the following contributes something to this objective:

1. Functional orientation
2. Choice of function names
3. Avoidance of ambiguity
4. Simple syntax
5. Intelligent defaults

Commands in the language correspond to functions the user needs to perform. The language is function-oriented rather than procedure-oriented; with this orientation it emphasizes user tasks rather than computer programming. Functional orientation, because of its close parallel to the mathematical concept of a function and its arguments, also provides a unity to the language. Compilers or interpreters for such languages are easy to build and maintain, or easy to change if necessary.

Names of functions were chosen to be obvious and intuitive. Functions have long and short versions of their names. The long version is suggestive of the functional operation and would normally be employed by a new user while he is gaining familiarity with the UC System. For example, the function, .in_file is used to define the user's current file to be searched by the .find function; the function, .out_file defines the user's current file into which records selected interactively out of the "hit list" from a .find function are saved. Short versions of function names could be used by an acceptably competent typist when entering sequences of commands (functions) from the alphanumeric keyboard of the terminal. Short versions would be memorized. For the above commands, the short versions are .ifl (.in_file), .of (.out_file), and .fi (.find).

Every effort was made to avoid function names that are confusing or ambiguous. For example, if a language has two commands -- SHOW and DISPLAY -- an occasional user may have a great deal of difficulty remembering which command does the operation he requires at the moment.

Language functions have an initial dot, which identifies them as keywords of the language and distinguishes them from parameters. Many functions of the

User Language require parameters. In the design of the language, every attempt was made to minimize the difficulty of entering commands. A command in the language consists of a function name followed by its parameters; in the simplest form of the language syntax, language elements (i.e. function names and parameters) are all separated by spaces. A more complex syntax is also available. Called the baseline syntax, it gives the experienced user more power, in that it supports composition of functions (i.e., functions can appear as arguments or parameters of other functions). The baseline syntax, which is used by the User Language interpreter and into which all language commands are translated, is presented in the history window of the text CRT during the execution of command sequences. This display makes constant learning reinforcement available, so that users who are interested may, without great difficulty, become accustomed to the more difficult syntax. Examples of the two forms of syntax are as follows:

Simple syntax -- .max WEIGHT1 WEIGHT2 WEIGHT3

Baseline syntax -- .max(WEIGHT1,WEIGHT2,WEIGHT3)

Certain functions in the language have parameters that are either static or remain constant over many uses of the function. In computer-based systems these are called "default parameters" or simply "defaults"; they need not be specified by the user when he enters the function. Two kinds of defaults frequently occur -- system defaults and user-specific defaults. Both kinds would be supplied by the UC System automatically at the time a function goes into execution. Parameter values for system defaults are obtained from tables in the language interpreter software. Values for user defaults are obtained from his profile table.

System defaults were well thought out and carefully chosen. They are both reasonable and intuitive and should minimize the chance of unexpected consequences. The baseline form of command sequences has parameters fully specified, so that a user is informed when defaults are being supplied and what their values are.

In addition to providing a language that is highly learnable in its form and organization, the UC System also promotes learning and familiarity by means of the .help function in the language. This function provides displays of tutorials of the UC System in general and of each user function:

.help (no parameters)   gives an introduction (at various levels of detail) to the system and the User Language.

.help .find   would display a description of the .find command, as well as instructions in its proper use.

6.2.1.2 Easy to Enter Commands Three input devices are provided to enter User Language functions: these are the Function Button keyboard, the Typewriter keyboard and the Interactive Menu Selection by means of the graphics display and the data tablet and stylus. Function buttons give two important capabilities in the UCS context. First of all, they provide a fast, efficient means of entering functions; this is especially important for non-typists who use the system. Secondly, frequently used parameters may be assigned to unused buttons on the keyboard. In this manner many functions can be entered entirely from the function buttons. The typewriter keyboard is used for entering numeric and textual parameters for functions. In certain modes of operation, fast typists may prefer to enter commands completely via the typewriter keyboard. Many or all of the language functions are also selectable as menu items on the graphics CRT. When a user's activity involves extensive interaction with data in the display window, command entry via menus may prove more convenient than moving to one of the other input devices.

Examples of language functions requiring few parameters or none have been given above. Other functions are highly parameterized by their nature; a good example is the .plot command. Some of the parameters of .plot are:

1. Scale of the x-coordinate;
2. Scale of the y-coordinate;
3. Whether the x-coordinate is linear, semilog or log;
4. Whether the y-coordinate is linear, semilog or log;
5. File containing the x-coordinate data to be plotted;
6. File containing the y-coordinate data to be plotted;
7. Labelling for the x-coordinate;
8. Labelling for the y-coordinate;
9. Labelling for the plot as a whole;
10. Erase or non-erase for the plot display window on the CRT (allows multiple, superimposed plots);
11. Dotted or solid line plot.

Additional help is made available to users in entering functions of this complexity. A system of fill-in forms prompts the user for each of the required parameters of the function, accepts the parametric data he enters into the form and routes the entire command to the language interpreter. Each function has a form associated with it. When, during the execution of the command string, function parameters are required, the form for a given function is displayed (temporarily) in the command history window. Forms are displayed as two-dimensional tables, with names for function parameters followed by fields into which the user is to input the parameter values. Where applicable, defaults are displayed in parameter fields. The user may type over a default, replacing it with a specified value if he so desires. Areas of the form outside of the parameter fields are protected by the terminal software.

6.2.1.3 Easy to Recover From Error Situations Several kinds of errors are possible in interaction with the System. In each case the UC System provides a meaningful error message or establishes an interaction by which a user can gain information to extract himself from the situation and avo l repeating the error.

6.2.1.4 Easy to Interact with Data Three of the principal kinds of data that analysts will work with are conventional text, record oriented text, and general record oriented data.

Conventional running text data are archived in text files, which are created, accessed and updated through the text editing subset of the User Language. Text is manipulated by means of its image on the text terminal CRT screen. Special character and word functions are available as function keys on the alphanumeric keyboard of the terminal, and are supported by software in the intelligent terminals. By this facility text data manipulation becomes simple and efficient.

Local data bases are implemented as files of structured text records in the UC System. In response to the .find (query) command, a hit file is produced and displayed on the display window of the graphics CRT. The user may interactively select individual records from the display by means of the data tablet and stylus (and the accompanying cursor generated on the CRT), causing these records to be saved in another file. If the display window can not contain the entire hit file, the user can cause the next set of records to be displayed by depressing the .continue button on the function button keyboard.

Data base records retrieved from remote DBM Systems are also displayed on the graphics CRT display window. Several User Language functions are provided to access specified fields of a data record.

6.2.1.5 Easy to `Program' for the Non-programmer Sequences of language functions followed by the .xeq function are executed together. This mode of operation gives the user the ability to minimize keystrokes by entering several functions on one line. It also allows related commands for a task to be combined. Carrying the method even farther, a file of User Language commands (called a UC Shell file) can be created beforehand and executed whenever desired by executing the .run_file function. For the breadboard UC language, this gives the user a kind of "macro" capability. In the complete implementation of the User Language a full capability for user defined functions have been supported.

User Language control structures have limited implementation in the breadboard UCS design. Part of the language support software, the UC Shell supports three kinds of capabilities that are not part of the User Language at present; these are branching instructions, iteration control instructions and conditional testing instructions. Users who have some background in programming would be able to construct UC Shell files that would use language commands in complex combinations. In the complete User Language implementation these control structure capabilities would also be added to the User Language itself.

6.2.1.6 Easy to Combine Capabilities and Access Distributed Resources By bringing together several functional areas into a single integrated system and a unified language, the UC System would have made it possible for FTD personnel to utilize computer support for analytical tasks in ways and to an extent not otherwise possible. From the User Language, facilities were to be provided to access data bases, perform calculations, do text and word processing, generate plots and utilize the programming resources available under the UNIX* system.

Communication with external hosts is an integral part of the UCS design concept. Commands are provided in the User Language to send files of control records to other computers in order to execute programs on those computers. In the full implementation of the language, commands were to generate the control files automatically. In addition, facilities are provided in the UC System to translate User Language queries to the query languages of DBM Systems residing on external host computers.

* UNIX is a Trademark of Bell Laboratories.

6.2.2 Other Language Design Considerations

6.2.2.1 Data Base and File Query Framework During the period in which the data base query functions in the User Language were being finalized, the UCS designers learned that FTD was moving toward increasing use of intelligence networks such as COINS and DIAOLS. A system called ADAPT had been developed under the sponsorship of Defense Advance Research Projects Agency and the Office of Naval Research. The first in a staged development series, ADAPT I, has the following features and capabilities:

1. Implementation on a PDP-11 computer under the UNIX operating system,

2. Support for interface to the the ARPANET,

3. A minimal user interface and language,

4. Implementation of a subset of a language called the Uniform Data Language (UDL),

5. Implementation of a language called the Data Definition Language (DDL),

6. Implementation of a language called the Transformation Definition Language (TDL),

7. Implementation (by means of DDL and TDL) of support for interface to four data base systems at remote hosts on the network and translation of queries in UDL to the respective query languages of the data base systems,

8. Implementation of support for translation of retrieved records from

network data bases to records in the UDL format.

The User Comm project team investigated the feasibility of utilizing some of the ADAPT capabilities -- particularly the DBMS query translation feature -- in the UC System. Although ADAPT was designed around a set of objectives (for example, network interface) that are different from the objectives of User Comm, the UCS designers concluded that it would be worthwhile to develop a remote data base retrieval capability around ADAPT. This is consistent with the basic User Comm philosophy of avoiding multiple user interfaces and languages at FTD wherever possible.

ADAPT software implements the UDL as a standard query capability, provides a mechanism for translating queries into DBMS query languages, and supports interrogation for some existing DBM Systems and display of retrieved records. It also has certain drawbacks:

1. Not all of UDL was implemented in ADAPT I. The DISPLAY command for viewing retrieved records is relatively limited. There is no support for report generation;

2. Local data bases are not supported;

3. UDL resembles st-ndard programming languages in its form and structure (for example it maintains the traditional distinction between expressions and statements). As a language, therefore, UDL is most appropriate to users with some background in programming.

In order to reconcile desirable ADAPT capabilities with the general User Comm approach, the UCS design embodies solutions to the above problems. UDL capabilities that were not implemented in ADAPT I are replaced by UCS capabilities that are more compatible with the User Language. Local files are supported, and query of these files is by the same language mechanisms that are used to query remote DBM Systems.

The basic User Language function for data base and file query is the .find command. Because of the complexity that may accompany a query having various conditions on a search, forms are used to assist in formulating queries. Through use of the form, parameters of the .find function are filled in interactively, with the system prompting or displaying the options at each point. The form support software thus acts as a preprocessor, turning the User Language query into a standard UDL query for processing by the software modules which were incorporated into the UCS from ADAPT I.

6.2.2.2 Language Access to Data Structures UDL requires the user to explicitly access required values within a record structure. There are three UDL commands that provide necessary iteration capability:

DO RECORD permits iteration through all records in the list created as a result of a query;

DO OCCURRENCE permits iteration through repeating groups within a record;

DO permits iteration through a sequence of commands based on a specified range value.

The approach taken in the User Language is to take the burden of finding desired fields or subfields of records off the user by making iteration over file records and fields an implicit operation.

6.2.2.3 Language Tunability Obtaining feedback from early users on the usability of the UCS and especially on the User Language was a long-term goal of its developers. Some of this feedback was intended to be obtained through post-use briefing sessions (i.e., a part of the Evaluation Program); however, the design for UCS software also provides for journaling of system/user interactions. Archived records of user sessions (with permission of the individual user) would indicate what commands cause difficulty, what aspects of language syntax cause problems, and what kinds of errors were made consistently. Analysis of this information would result in data useful for adjustment of the UC System and the User Language in the direction of improved user reaction and success.

6.3 The Minicomputer System

The concept for an FTD User Communications facility has evolved around the notion of a hierarchy of processors, and we believe that this concept is still basically sound. It falls in line with modern notions of modularity and separation of function, from both software and hardware points of view. In addition to the notion, discussed elsewhere, of offloading the host machine, the distributed system greatly simplifies development. It is difficult to justify experimental procedures on a large, costly, mainframe, while it is reasonable to do breadboarding of hardware and software on a dedicated minicomputer. Once such a system has been finalized, checked out, and measured, it can be replicated. Adding more users requires adding more hardware, but requires very little additional development, and involves no degradation of response time for local functions.

In preparing for the User Communications project, it was necessary to judge and select the hardware complement which would satisfy the system requirements. Having dealt with and designed around that hardware complement, we are now in a better position to judge and evaluate the choices made.

Digital Equipment Corporation (DEC) has long been the leader in the minicomputer field. They have a widespread, well-trained field service department. Their PDP 11 series computers are extremely popular. We have found their equipment to be well designed relative to other vendors in the field. A wide range of peripheral equipment is available, both from DEC and from independent vendors. DEC's large customer base has also lead to the availability of a considerable amount of software for the PDP 11; in particular, the availability of the UNIX operating system and its related software is very significant. One other significant feature of the PDP 11 is its long lifetime. All indications are that DEC is committed to very long term support of the machine, and the availability of compatible software and hardware is increasing.

The choice of the particular model of PDP 11, while not as clear, was, we feel, correct. The PDP 11/70 has several significant features lacking in smaller models.

The most important feature of the PDP 11/70 is support of a large amount of main memory. This is particularly important in a multi-user, time-sharing environment, where there may be many processes, as well as the operating system itself, competing for memory. Less memory means more disk swapping, resulting in drastic degradation of response time.

The PDP 11/70 also offers separated data and instruction space, a mechanism which can provide a single process with an address space of up to 64K words.

Increased address space reduces the need for overlays and temporary disk files, and makes feasible certain programs such as LISP interpreters.

Other outstanding features of the PDP 11/70 are the cache and the massbus. The cache, through sophisticated logic, provides significantly higher memory speeds when operating under appropriate conditions. The massbus provides a 32-bit data path for high-speed devices such as disk and magtape. As an interim measure, the User Communications project utilized a disk controller which interfaced to the PDP 11/70's unibus, rather than its massbus. We found this to be an unsatisfactory arrangement; the heavy load on the unibus interfered with the functioning of other devices, disk throughput was affected by use of the slower bus, and the utility of the cache was greatly undermined by heavy DMA (direct memory access) activity.

Generally, we found that certain problems arise in multi-source systems. While purchasing hardware from several vendors provides flexibility in selecting the design most appropriate to the task, as well as reducing initial costs where competitive products are available, maintenance became more complicated and mean time to repair increased. Pinpointing the source of the problem can be difficult, and it may be necessary to call in field service personnel from more than one vendor, with finger-pointing often resulting. These factors should be carefully considered during initial hardware evaluation.

Transferring I/O functions to a subordinate processor is a natural extension of the distributed processing philosophy which led to the User Communications System concept of offloading functions from the FTD host machine onto a minicomputer (viz. the PDP 11/70). Offloading the PDP 11/70 in turn frees it to perform more extensive local functions, and allows the use of a human-oriented, non-real-time operating system such as UNIX. Also, multiple user work-stations can be added by adding subordinate processors to a single PDP 11/70, with minimal cost and reduced degradation of system response time.

The PDP 11/04 was an appropriate choice as a subordinate I/O processor for several reasons. The PDP 11/04 is a small but powerful unbundled processor, well suited to a cost effective treatment of the task. Since the PDP 11/04 is a member of the same family as the PDP 11/70, there is a high degree of software compatibility, and almost complete hardware compatibility between the two machines. Programs and peripherals can be developed and tested on the larger machine, and then transferred to the smaller machine. Since the PDP 11/70 and the PDP 11/04 have the same bus structure, high-speed interfaces are available from the vendor.

One of the difficulties with a distributed system of this sort is in communication between the processors. Since the subordinate processor handles several different devices for two different user work-stations, the communications link can be a significant logical bottleneck. The DMC11 link used in the UCS was chosen for its speed and its downloading capabilities; however, it is by nature a single serial link and it requires complicated driving software to utilize it fully. A possibly more effective solution might be a core window (e.g. DA11-F) together with a downloading mechanism; this alternative requires careful investigation.

The complement of work-station peripherals selected for the UCS provided a full range of facilities. The programmable alphanumeric CRT terminal provides a keyboard and a display area with highlighting features such as blinking and reverse video, with local editing and scrolling capabilities not available in non-programmable terminals. The raster graphics display processor provides a selectively erasable and scrollable area for display of text and/or graphic materials, and can be expanded to provide color or grayscale. The function keyboard and the data tablet provide alternate input facilities which can reduce the amount of typing and the number of input errors. The floppy disk drives provide a low cost medium for storage of individual user data; in conjunction with UNIX, the floppies provide a private structured file facility.

One weakness of the work-station hardware is the difficulty of providing visual feedback of the position of the data tablet stylus relative to the display. Producing a cursor on the graphics screen via software puts a heavy interrupt load on the processor, and can result in jerky movement of the cursor. Hardware configurations which automatically produce a smooth cursor are possible, and should be considered.

The Versatec printer/plotter proved to be quite satisfactory as a hardcopy output device. The 8.5" x 11" paper is more convenient than the standard 11" x 17" computer output, and is better suited to document production. Plot mode allows interspersal of text and graphics. The 200 bit per inch resolution allows soft printing (as opposed to the hardware character set) with quite high quality. Variable width fonts and intermixing of bold, italic, and roman text are possible. While not the quality of a phototypesetter, the Versatec is significantly faster and less expensive.

Generally, we have found that the distributed minicomputer system is an appropriate way to address the problem of a human-engineered user communications interface. As technological advances are made and production costs drop, this approach will become even more attractive.

6.4 The UNIX Operating System

During the contract period, the User Communications project team had the opportunity to use and observe the UNIX operating system, and to evaluate it in terms of initial expectations.

The features of UNIX which made it attractive initially have generally reached or exceeded expectations. The system has proven to be an excellent developmental tool for both programs and documentation.

The screen editor makes text entry and modification simple and pleasant, and allows clerical personnel to learn its use quickly, with a minimum of assistance.

The NROFF text formatting program permits rapid production of documents without extensive retyping efforts. This, together with an extensive set of tools for operations such as text searching, file comparison, and spelling error detection, results in an environment highly conducive to on-line maintenance of program documentation, reports, design documents, and any other form of text.

The use of the high-level structured programming language C, together with a powerful set of library subroutines, provided an excellent facility for the development of utility and statistical programs produced during the project effort. Programmers on the project who had had no previous experiences with C found it easy to learn, and the structured facilities made programs easier to read, modify, and debug than assembly language or FORTRAN. Also, since the vast majority of the code in the operating system and its accompanying programs is written in C, reading and understanding these programs was greatly eased.

The relative simplicity of the UNIX I/O mechanism, together with the use of C, makes I/O drivers fairly easy to write, although some of the more obscure details are not documented and must be extracted by reading existing code.

Sysgen, backup, and other procedures are made simple and reliable through the UNIX command interpreter ("shell") and the MAKE program. Once a new driver is coded, it can be added to the system in less than five minutes. Adding a new device for which a driver already exists consists of adding one line to a certain file, executing one command, and booting the new system.

The project team feels that the decision to procure UNIX as a supported product from Interactive Systems Corporation (ISC) was a wise one. The product (Interactive System/Workbench) is based upon the Programmer's Workbench version of UNIX, and has considerably more functionality than the standard version. ISC has expended considerable effort to establish uniformity and consistency throughout the system, making it an even more useful tool. ISC's physical proximity and cooperative attitude also helped in obtaining needed consultation and advice. As one of ISC's first customers, OSI watched their product and their approach mature, and we would advise seriously considering them when procuring future UNIX systems.

One of the major concerns which we had when initially evaluating UNIX was reliability. We found, however, that the system, as improved under Programmer's Workbench and by ISC, is extremely stable. We never had a system crash attributable to software. Although we had many hardware crashes, we found that by performing automatic disk updates frequently, we avoided lost or corrupted files.

The UNIX utility and systems programs, while not completely error free, seem unusually stable as compared to other systems we have used. Most bugs encountered are well-behaved (i.e. consistent) and occur in seldom-used features. We attribute this level of reliability partly to the use of a structured high-level language throughout the system, partly to its modular design, and partly to the high caliber of people who have developed the system.

Some of the known or suspected drawbacks have been borne out, and some problems were encountered which were not foreseen, or the significance of which had been underestimated.

Documentation for UNIX has always been a problem. Although most of the information necessary is available, we found that locating the right information at the right time was quite difficult. The sheer mass of detail associated with the system was often found to be an obstacle. We found that we were sometimes using complicated techniques to solve problems for which UNIX provides simple solutions, because the available documentation was purely descriptive and had not led us in the right direction. Those individuals who spent considerable time studying the documentation and experimenting became valuable sources of information, and their time was further taken up as consultants. We feel that UNIX would greatly benefit from documentation with better human engineering: examples, scenarios, and discussions of the purpose or intent of various facilities.

One of UNIX's failings is the lack of a good general-purpose interprocess communication facility. Although the pipe mechanism is an excellent tool, it is overly restrictive. A shared memory facility or named pipes would have allowed for a more efficient UCS design. The UNIX signal mechanism has serious drawbacks as a software interrupt mechanism, and seems not to have been intended for that purpose.

The UNIX I/O system, while simple and efficient, poses serious problems for the systems designer. The lack of asynchronous I/O in some cases produces inefficient implementations, and in others necessitates complicated I/O drivers. More severe is the lack of queueing or bracketing facilities; if two processes write to the same device concurrently, the outputs generally are mixed. Individual drivers can be written to perform their own bracketing, but they can become locked up if a signal occurs. The system can probably be modified to prevent this, and ISC has looked at the problem.

One of the problems of UNIX for the UCS is the unavailability of a core-only system for the PDP 11/04. Whereas DEC's RSX family of systems includes RSX11S, no comparable system is available with UNIX. RSX11S requires an RSX11M or RSX11D system for development and maintenance. A mini-UNIX is available, but it is intended for single-user interactive use, and requires a disk drive. Although stand-alone C programs could be written and downloaded into the PDP 11/04, the C compiler would have to be modified to use only the restricted instruction set of the smaller machine.

UNIX, like any other operating system, has both strong and weak points. It was designed as a timesharing system to provide useful services to on-line users, and in this function we have found it to perform superbly. Applications software has extended basic UNIX in a number of directions; two of the more important areas are word processing and software development.

In a sense the User Comm project has participated in the "coming of age" of UNIX. During this period UNIX has come out of the laboratories and has become a workhorse system in government and private industry. Documentation has improved, support has become available and mechanisms have been introduced to provide control and standardization. UNIX is now being introduced on a number of commercial computers outside of the Digital PDP 11 line. The ADAPT System discussed in Section 6.2 was developed by an industrial contractor under UNIX for important segments of the U.S. intelligence community.

The User Communications project team feels that, despite its flaws, UNIX is an excellent developmental tool, and contributes a great deal to a bread-board system concerned with human engineering and communications.

6.5 Overview of the User Communications System Design

6.5.1 UCS Logical Configuration In simplest terms, the User Communications System can be thought of as a centralized user command handler which communicates with a number of functional modules so that each module executes a user command or group of commands. This logical organization is shown in Figure 6.5-1.

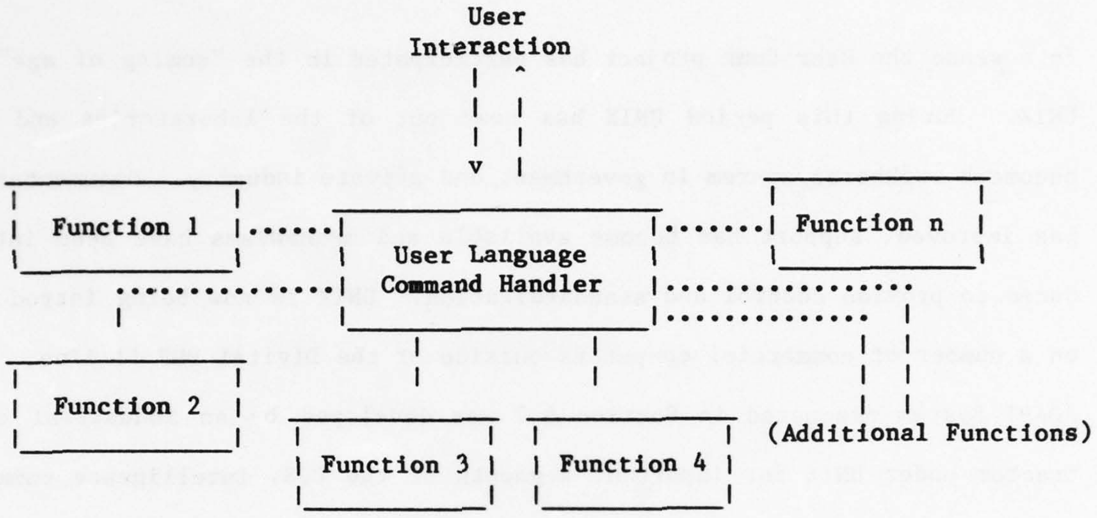


Figure 6.5-1. Logical Organization of the UCS

The Command Handler is a set of program modules which accept User Language commands from various devices, route each command to one of the functional modules for execution and display results to the user.

6.5.2 UCS Physical Configuration The UCS is controlled by a PDP 11/70 minicomputer. The essentials of the UCS reside on the PDP 11/70 and the PDP 11/04. Access to applications programs and/or data bases is provided by means of a link to one or more larger mainframe hosts. In the breadboard system, the user interfaces with the UCS at one of two workstations; the workstation configuration and the physical interfaces will be detailed in the next section. The UCS utilizes a 200 megabyte disk for local data storage, and a magnetic tape unit for archiving of data and backup of system and user files. Communication between the Mainframe Host and the PDP 11/70 is effected via a yet unspecified DMA interface. The system operator manages the system through the DECwriter system console. Hardcopy may be produced on the Versatec printer/plotter. Communication between the 11/70 and 11/04 processors is provided by means of a DMC-11 interface for each processor. Attached to the PDP 11/04 by means of a Unibus* interface is the Genisco Programmable Graphics Processor. The PDP-11/04 passes data and programs to the graphics processor; in doing so, it offloads a heavy I/O activity from the 11/70 processor. The UCS physical organization is shown in Figure 6.5-2.

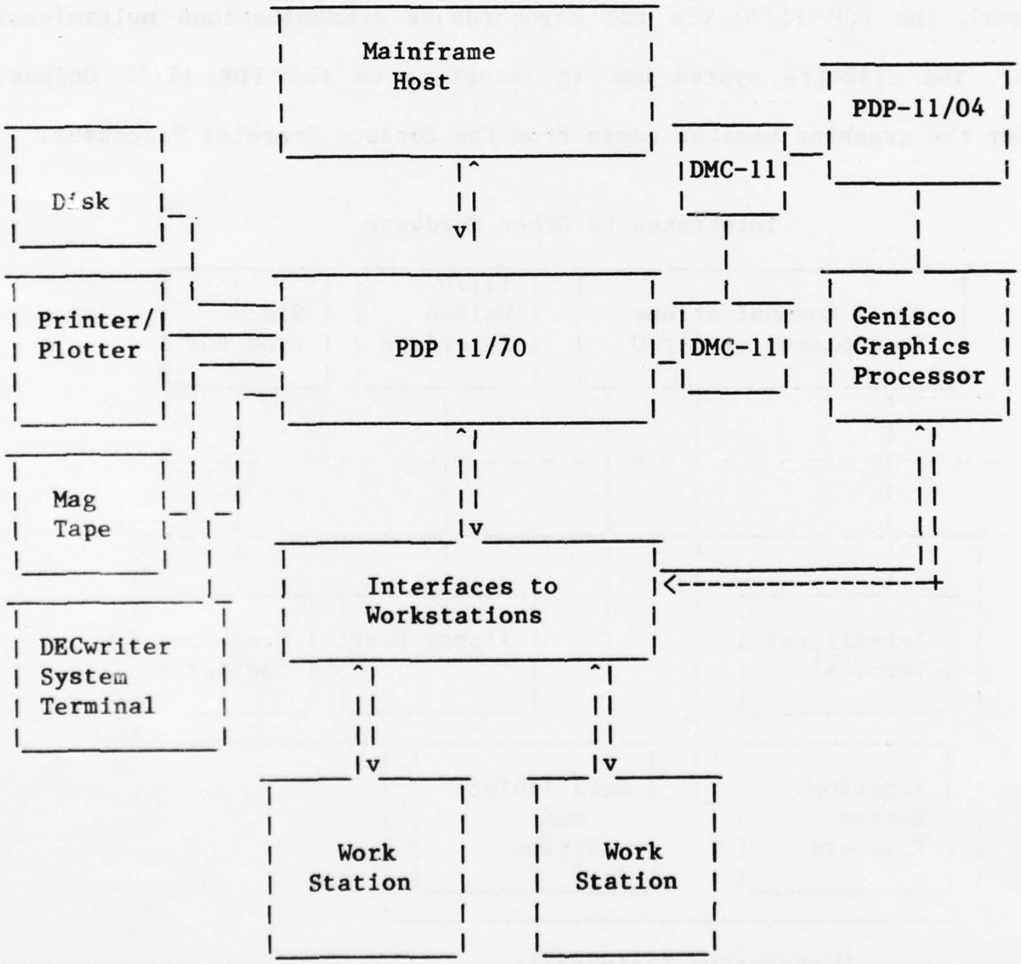


Figure 6.5-2. Physical Organization of the UCS

* UNIBUS is a registered trademark of Digital Equipment Corporation.

6.5.3 UCS Workstation Organization Each workstation contains an intelligent terminal with its own alpha-numeric keyboard, a graphics monitor, a data tablet and stylus for interaction with the graphics system, a function button keyboard for entering commands and a diskette (floppy) disk system with two drives.

As shown in Figure 6.5-3, there are three kinds of physical interface between a workstation and the other UCS hardware. The intelligent terminal, the graphics data tablet and stylus and the function button keyboard communicate with the PDP 11/70 via the asynchronous communications multiplexor (DZ-11). The diskette system has an interface on the PDP 11/70 Unibus. Video for the graphics monitor comes from the Genisco Graphics Processor.

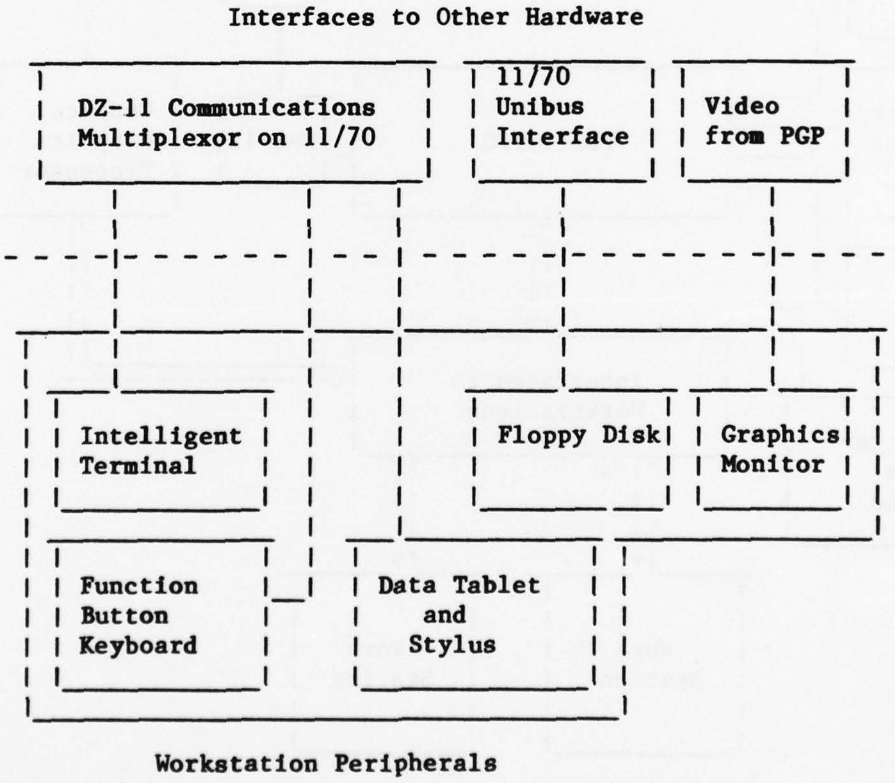


Figure 6.5-3. Workstation Peripherals and Interfaces

6.5.4 UCS Design Solutions Although detailed design of the UCS is not complete, most of the important problems for an initial breadboard system have been solved. Some of these solutions are presented here.

6.5.4.1 Terminal Keyboard Conventions The problem is that UNIX shares with other operating systems* the characteristic of enforcing character conventions for terminal keys and functions that may conflict with conventions of applications programs that are highly terminal-oriented; for example there are different character sequences employed for erase or backspace (i.e., Control (CTRL) h) in dialog with the UNIX command interpreter than are used in the CRT screen editor program (the BACKSPACE key for certain purposes and the DELETE CHARACTER function key for others). These differences can be very confusing, especially to new users.

The solution arrived at in the UC System is to establish a set of keyboard conventions that are identical to or compatible with those of the screen editing functions, and to enforce these conventions over all functions in the UC mode of operation. In other words, most UCS users will be unaware of UNIX conventions. Software in the intelligent terminal will scan the characters typed in and make a local interpretation, where appropriate, instead of passing the character on to UNIX terminal handling software.

6.5.4.2 UCS Initialization Sequence The problem is that, whereas a user under UNIX obtains access to the computer system by means of a UNIX login procedure, UCS needs to have its own control over user access. The UC System is more extensive than just UNIX; it is a distributed system in which

* This "problem" in the UCS design and those that follow are not a consequence of a deficiencies in UNIX, but rather reflect the nature of operating systems in general; where "UNIX" is mentioned here, one could just as accurately say, "an operating system".

processors other than the PDP-11/70 must be set up to cooperate in the various functions a UCS user will execute.

The solution is that the UNIX system initialization procedure is modified, so that when UNIX is booted as a User Comm System, special UNIX processes initialize the users' workstation devices and start up the system.

1. Software is downloaded into the intelligent terminal;

2. Software is downloaded into the Programmable Graphics Processor;

3. Terminal mode tables used by UNIX system processes are set to reflect UCS conventions;

4. System processes are started up which activate the graphics tablet and stylus so that they will begin responding to user inputs;

5. Communication paths are set up between the function button keyboard and the intelligent terminal so that character string names of User Language commands are echoed on the terminal CRT when a function button is depressed;

6. Interface hardware or software protocols for links to external processors are initialized.

7. Control is passed to a process like the UNIX Shell (Command Interpreter) which cooperates with the intelligent terminal software in the execution of UCS commands.

6.5.4.3 Multiple CRT Windows The problem is that UNIX, without extensive modification of the terminal handling software, can not easily maintain distinct windows on a terminal CRT screen. It is thus not practical to attempt multiple window displays using only what UNIX provides.

The solution is to incorporate windowing software in the terminal itself. With some modification of the UNIX terminal driver software, message streams can be identified as coming from a certain process running under UNIX, and synchronization techniques can be utilized to keep the message streams distinct. When identified character streams arrive in the terminal, they are routed to a terminal routine which causes them to be displayed at a prescribed position in one of the terminal windows.

6.5.4.4 UNIX Escape Mode The problem is that computer-experienced UCS users who want to take advantage of the sophisticated programming resources available under UNIX will have UNIX hidden from them. This is because the User Comm System takes over the machine and initializes things in its own way.

The solution is to provide in the User Language a .unix command, which reinitializes the system in a UNIX mode. The terminal reverts to the kind of "dumb" terminal UNIX expects, and the standard UNIX command interpreter is given control for that user.

6.5.4.5 Simulation of Terminals on Host Computers The problem is that when the UC System is interfaced to a host computer by means of a terminal interface, the host must see the UCS as just another terminal. In this case the workstation terminal has to simulate the characteristics of a host terminal (e.g., a Univac Uniscope 200).

The solution is to provide a User Language command for initiating a terminal link to a given host. After the link is completed, and contact has been established between the two systems, special software is downloaded into the intelligent terminal that will enable it to simulate the protocols of terminal on the host system.

REFERENCES

Operating Systems, Inc., STIS User Communications Interface: Functional Description, RADC Report R-75-028, 31 December 1975.

Planning Research Corporation, FTD User Communications Interface: Functional Description, System/Subsystem, Submitted under RADC Contract No. F30602-75-0345, February 1977.

Planning Research Corporation, FTD User Communications Interface: System/Subsystem Specification (External Subsystem Report), Submitted under RADC Contract No. F30602-75-C-0345, December 1976.

Planning Research Corporation, FTD User Communications Interface, Program Specification, Submitted under RADC Contract No. F30602-75-C-0345, February 1977.

Operating Systems, Inc., FTD User Communications Interface - Final Technical Report, Submitted under RADC Contract F30602-75-C-0345, Subcontract D-1021 to PRC/Information Sciences Co., May 23, 1977.

Operating Systems, Inc., Proposal for FTD User Communications System Development, Submitted to RADC 4 May, 1977.

Foreign Technology Division, Scientific and Technical Information System (STIS) -- STI System User's Guide.

Planning Research Corporation, Data Base Support for the Implementation of Intelligence Production System. Interim Report Submitted under RADC Contract No. F30602-73-0391, 1 July 1972.

Operating Systems, Inc., A Comparison of the UNIX and RSX-11M Interactive PDP-11 Operating Systems for FTD User Communications Development, October 1977.

Operating Systems, Inc. FTD User Communications Interface: Functional Description OSI: R78-004 Submitted under RADC Contract No. F30602-77-0220, March 1978.

Rosenthal, H, et. al., Auerbach Associates Inc. (Calculon Corp.), STIS Interface Requirements to IPL, FORTRAN, and User Comm, Report AAI-2432-TR-1, Submitted to Operating Systems, Inc. under Subcontract No. 77-002; Prime--RADC Contract No. F30602-C-77-0220, December, 1977, Revision; June, 1978.

Operating Systems, Inc., FTD User Communications System: System/Subsystem Specification (DRAFT), RADC Contract No. F30602-C-0220, June, 1978.

Operating Systems, Inc., "Preliminary Analysis of the Information Activities Pattern Survey Questionnaire Taken at FTD in February 1978" (DRAFT), RADC Contract No. F30602-C-0220, September, 1978.

Logicon, Inc., ADAPT I Uniform Data Language (UDL): A Final Specification, Final Technical Report: Report 76-C-0899-1, submitted to Defense Advanced Research Projects Agency and Office of Naval Research, Informations Systems Program, January 1978.

Logicon, Inc. ADAPT I Final Functional and System Design Specification. Report 76-C-0889-2.

Operating Systems, Inc., FTD User Communications Requirements Analysis, Appendix to FTD User Communications System: Functional Description, March, 1978.

Bell Laboratories, Inc., Ed., PWB/UNIX User's Manual, May 1977.

This document was produced on the User Communications PDP-11/70 computer using the text and document handling facilities provided under the UNIX operating system.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.